Amendments of the Claims:

A detailed listing of all claims in the application is presented below. This listing of claims will replace all prior versions, and listings, of claims in the application. All claims being currently amended are submitted with markings to indicate the changes that have been made relative to immediate prior version of the claims. The changes in any amended claim are being shown by strikethrough (for deleted matter) or underlined (for added matter).

Claims 1-14. (Canceled)

15. (Currently Amended) A practical numerical computer-implemented method for reliably computing a dynamical decomposition point of a stable equilibrium point for large-scale nonlinear systems, comprising the steps of:

a) given a stable equilibrium point x_s;

b) moving along a search path $\varphi_t(x_s) \equiv \left\{ x_s + t \times \hat{s}, \quad t \in \Re^+ \right\}$ starting from x_s and detecting an exit point, x_{ex}, at which said search path $\varphi_t(x_s)$ exits a stability boundary of a stable equilibrium point x_s;

c) using said exit point x_{ex} as an initial condition and integrating a nonlinear system (4.2) to an equilibrium point x_d; and

d) computing said dynamical decomposition point with respect to the stable equilibrium point x_s wherein said search path is x_d.; and

e) displaying the dynamical decomposition point.

16. (Currently Amended) The method of claim 15, wherein a method for computing said exit point of the nonlinear system (4.3) comprises the step of moving along said search path $\varphi_t(x_s) \equiv \left\{ x_s + t \times \hat{s}, \quad t \in \Re^+ \right\}$ starting from x_s and detecting said exit point x_{ex}, which is a first local maximum of an objective function $C(x)$ along said search path $\varphi_t(x_s)$.

2

17. (Currently Amended) The method of claim 15, wherein a method for computing a dynamical
    decomposition point comprises the steps of:

    a) using said exit point x_{ex} as an initial condition and integrating a nonlinear
       system ~~(4.2)~~ to a first local minimum of a norm $\|F(x)\|$ along the
       corresponding trajectory, where $F(x)$ is a vector field of <u>the nonlinear</u>
       <u>system</u> ~~(4.2)~~, and letting the point at which the first local minimum of
       $\|F(x)\|$ occurs be denoted x_d^0, and is called the minimum distance point
       (MDP); and

    b) using said MDP x_d^0 as an initial guess and solving a set of nonlinear algebraic
       equations of said vector field ~~(4.2)~~ $F(x) = 0$, wherein a solution is x_d, and
       a dynamical decomposition point with respect to the local optimal solution
       x_s and said search path $\varphi_t(x_s)$ is x_d.

18. (Currently Amended) The method of claim 15, wherein a method for computing said exit
    point with respect to a stable equilibrium point of the nonlinear system ~~(4.2)~~ and a search
    vector comprises the step of computing an inner-product of said search vector and the
    vector field of <u>the nonlinear</u> system ~~(4.3)~~ at each time step, by moving along said search
    path $\varphi_t(x_s) \equiv \{x_s + t \times \hat{s}, \quad t \in \Re^+\}$ starting from x_s and at each time-step, computing an
    inner-product of said search vector $\hat{s}$ and vector field $F(x)$, such that when a sign of
    said inner-product changes from positive to negative, said exit point is detected.

19. (Currently Amended) The method of claim 15, wherein a method for computing said exit
    point of <u>the</u> nonlinear system ~~(4.3)~~ with respect to a stable equilibrium point and a search
    vector comprises the steps of:

    a) moving from said stable equilibrium point along said search vector until an
       inner-product of said search vector and the vector field of <u>the nonlinear</u>
       system ~~(4.3)~~ changes sign between an interval $[t_1, t_2]$;

3

b) applying a linear interpolation to an interval $[t_1, t_2]$, which produces an intermediate time t_0 where an interpolated inner-product is expected to be zero;

c) computing an exact inner-product at t_0, such that if said value is smaller than a threshold value, said exit point is obtained; and

d) if said inner-product is positive, then replacing t_1 with t_0, and otherwise replacing t_2 with t_0 and going to step b).

20. (Currently Amended) The method of claim 15, wherein a method for computing a minimum distance point (MDP) of the nonlinear system (4.2) satisfying required conditions (C1) and (C2) comprises the steps of:

a) using said exit point as an initial condition and integrating the nonlinear system (4.2) for a few time-steps, and letting the end point be denoted as the current exit point;

b) checking convergence criterion, and, if a norm of said current exit point obtained in step a) is smaller than a threshold value, then declaring said point as said MDP and stopping the process, otherwise, going to step c); and

c) drawing a ray connecting a current exit point on a trajectory and a local optimal solution (equivalently, a stable equilibrium point), replacing said current exit point with a corrected exit point, which is a first local maximal point of objective function along said ray, starting from the stable equilibrium point, and assigning this point to said exit point and going to step a).

21. (Currently Amended) The method of claim 15, wherein a method for computing said dynamical decomposition point of the nonlinear system (4.2) satisfying required conditions (C1) and (C2) with respect to a stable equilibrium point x_s and a search vector $\hat{s}$, comprises the steps of:

4

a) moving along said search path $\varphi_t(x_s) \equiv \left\{ x_s + t \times \hat{s}, t \in \Re^+ \right\}$ starting from x_s and detecting a moment that an inner-product of said search vector $\hat{s}$ and the vector field $F(x)$ of ~~the~~ nonlinear system ~~(4.2)~~ changes sign, between an interval $[t_1, t_2]$, stopping this step if t_1 is greater than a threshold value and reporting that there is no adjacent local optimal solution along this search path, otherwise, going to step b);

b) applying linear interpolation to said interval $[t_1, t_2]$, which produces an intermediate time t_0 where said interpolated inner-product is expected to be zero, computing an exact inner-product at t_0, and if said value is smaller than a threshold value, said exit point is obtained, and going to step d);

c) if said inner-product is positive, then replacing t_1 with t_0, and otherwise replacing t_2 with t_0 and going to step b);

d) using said exit point as an initial condition and integrating ~~a~~ the nonlinear system ~~(4.2)~~ for a few time-steps, and letting the end point be denoted as the current exit point;

e) checking convergence criterion, and if a norm of said point obtained in step d) is smaller than a threshold value, then declaring said point as the MDP and going to step g), otherwise going to step e);

f) drawing a ray connecting a current exit point on said trajectory and a stable equilibrium point, replacing said current exit point with a corrected exit point which is a first local maximal point of objective function along said ray starting from the stable equilibrium point, and assigning this point to said exit point and going to step d); and

g) using said MDP as an initial guess and solving a set of nonlinear algebraic equations of the vector field ~~of the nonlinear system~~ ~~(4.2)~~ $F(x) = 0$,

5

wherein a solution is t_d, such that said DDP with respect to a stable

equilibrium point x_s and search vector $\hat{s}$ is x_d.

22. (Cancelled)

23. (Currently Amended) A hybrid search computer-implemented method for obtaining a local

optimal solution of a general unconstrained nonlinear programming problem ~~(4.1)~~

starting from any initial point, comprising the steps of:

a) given an initial point x_0;

b) integrating a nonlinear dynamical system ~~described by (4.2)~~ that satisfies

required conditions ~~(C1) and (C2)~~ from said initial point x_0 to obtain a

trajectory $\phi_t(x_0)$ for n time-steps, n being an integer, and recording the

last time-step point as the end point, and if it converges to a local optimal

solution, then stopping, otherwise, going to step (c);

c) monitoring a desired convergence performance criterion along the trajectory

$\phi_t(x_0)$ in terms of the rate of decreasing values in the objective function

under study, and if the desired criterion is satisfied, then using the end

point of trajectory $\phi_t(x_0)$ as the initial point and going to step (b),

otherwise, going to step (d); and

d) applying an effective local optimizer (*i.e.*, a method to find a local optimal

solution) from said end point in step (b) to continue the search process,

and if it finds a local optimal solution, then displaying the local optimal

solution and stopping, otherwise, setting the end point of trajectory $\phi_t(x_0)$

as the initial point, namely x_0, and going to step (b).

24. (Cancelled)

6

25. (Previously Presented) The method of claim ~~14~~23, wherein in step (a) the initial point x_0 is given by a method comprising the steps of:

    a) moving along a search path starting from a local optimal solution x_{opt} and applying a DDP search method to compute a corresponding DDP, and if a DDP can be found, then going to step (b), otherwise, trying another search path and repeating this step;

    b) letting said DDP be denoted as x_d, and if x_d has previously been found, then going to step a), otherwise going to step c); and

    c) setting $x_o = x_{opt} + (1 + \varepsilon)(x_d - x_{opt})$ where ε is a small number, and applying a hybrid search method starting from x_0 to find a corresponding adjacent local optimal solution.

26. (Previously Presented) A computer-implemented method for obtaining a global optimal solution of unconstrained nonlinear optimization problems, comprising the steps of:

    a) choosing a starting point;

    b) applying the hybrid search method of claim 23 using said starting point to find a local optimal solution x_s^0;

    c) setting $j = 0$, $V_s = \{x_s^0\}$, $V_{new}^j = \{x_s^0\}$ and $V_d = \{\phi\}$;

    d) wherein set $V_{new}^{j+1} = \{\phi\}$ and for each local optimal solution in V_{new}^j (i.e., x_s^j), performing steps (e) through (k);

    e) defining a set of search vectors S_i^j, $i = 1,2, ..., m_j$, and setting $i = 1$;

    f) if $i > m_j$, then going to step (l); otherwise, applying a DDP search method along the search vector S_i^j to find a corresponding dynamical

7

decomposition point (DDP), and if a DDP is found, then going to step (g), otherwise, setting $i = i + 1$ and going to step (f);

g) letting the found DDP be denoted as $x_{d,i}^j$, checking whether it belongs to the set V_d, i.e., $x_{d,i}^j \in V_d$?, and if it does, then setting $i = i + 1$ and going to step (f), otherwise setting $V_d = V_d \cup \left\{ x_{d,i}^j \right\}$ and going to step (h);

h) for the DDP $x_{d,i}^j$, performing steps (i) through (j) to find a corresponding adjacent local optimal solution;

i) letting $x_{0,i}^j = x_s^j + (1 + \varepsilon)\left(x_{d,i}^j - x_s^j \right)$, where ε is a small number;

j) applying said hybrid search method using $x_{0,i}^j$ as the initial condition to find the corresponding local optimal solution, and letting it be denoted as $x_{s,j}^i$;

k) checking whether $x_{s,j}^i$ has been found before, i.e., $x_{s,j}^i \in V_s$?, and if it has already been found, then setting $i = i + 1$ and going to step (f), otherwise, setting $V_s = V_s \cup \left\{ x_{s,j}^i \right\}$ and $V_{new}^{j+1} = V_{new}^{j+1} \cup \left\{ x_{s,j}^i \right\}$ and setting $i = i + 1$ and going to step (f);

l) examining the set of all newly computed local optimal solutions, V_{new}^{j+1}, and if V_{new}^{j+1} is non-empty, then setting $j = j + 1$ and proceeding to step (d), otherwise proceeding to the next step; ~~and~~

m) identifying the global optimal solution from said set of local optimal solutions V_s by comparing their corresponding objective function values in set V_s and;

n) displaying the global optimal solution.

27. (Cancelled)

8

28. (Currently Amended) A computer-implemented method for obtaining a global optimal

solution of a constrained nonlinear programming problem ~~(4.5)~~, comprising the steps of:

a) Phase I: finding all feasible components of the constrained nonlinear

programming problem wherein one effective local method is combined

with said dynamical trajectory method, comprising the following steps to

find a feasible solution of the constrained optimization problem ~~(4.7)~~:

i) given an initial point x_0;

ii) integrating a nonlinear dynamical system ~~described by (4.9)~~ that

satisfies <u>required</u> conditions ~~(C1-1) and (C1-2)~~ from said initial

point to obtain a trajectory $\phi_t(x_0)$ for n time-steps, n is an integer,

and recording the last time-step point as the end point, and if it

converges to a feasible solution, then stopping, otherwise, going to

step (iii);

iii) monitoring a desired convergence performance criterion in the

objective function $\|H(x)\|$, a vector norm of $H(x)$ in ~~(4.7)~~ <u>the

constrained optimization problem</u>, and if the desired criterion is

satisfied, then using the end point of trajectory $\phi_t(x_0)$ as the initial

point and going to step (ii), otherwise, going to step (iv); and

iv) applying an effective method from the said end point in step (b) to

continue the search process, and if it finds a feasible solution of

~~(4.7)~~ <u>the constrained optimization problem</u> then stopping,

otherwise, setting the end point of trajectory $\phi_t(x_0)$ in step (ii) as

the initial point and going to step (ii) ; and

b) Phase II: finding all local optimal solutions for the constrained nonlinear

programming problem in each feasible component;

9

c) choosing a global optimal solution for the constrained nonlinear programming problem from the local optimal solutions found in step (b); and

d) displaying the global optimal solution for the constrained nonlinear programming problem.

29. (Currently Amended) The method of claim 28, comprising the steps of:

a) approaching a path-connected feasible component of a constrained optimization problem (4.5); and

b) escaping from said path-connected feasible component and approaching another path-connected feasible component of said constrained optimization problem (4.5).

30. (Currently Amended) The method of claim 28, comprising the steps of:

a) approaching a stable equilibrium manifold of a nonlinear dynamical system (4.9) satisfying required conditions (C1-1) and (C1-2); and

b) escaping from said stable equilibrium manifold and approaching another stable equilibrium manifold of said nonlinear dynamical system (4.9) satisfying said conditions (C1-1) and (C1-2).

31. (Currently Amended) The method of claim 28, comprising the steps of:

a) starting from a point in a feasible component and approaching a local optimal solution located in said feasible component of an optimization problem (4.7); and

b) escaping from said local optimal solution and approaching another local optimal solution of said feasible component of said optimization problem (4.7).

32. (Currently Amended) The method of claim 28, comprising the steps of:

10

a) in a deterministic manner, first finding all stable equilibrium manifolds of a

nonlinear dynamical system that satisfies required conditions-(C1-1) and

(C1-2);

b) in a deterministic manner, finding all stable equilibrium points of a nonlinear

dynamical system that satisfies said conditions-(C2-1) and (C2-2); and

c) then from said stable equilibrium point, finding a global optimal solution.

33. (Currently Amended) The method of claim 32, comprising the steps of:

a) given a feasible solution of constrained nonlinear programming problem-(4.5);

b) finding a stable equilibrium point of a nonlinear dynamical system (4-10)

satisfying required conditions-(C2-1) and (C2-2);

c) moving from said stable equilibrium point to a dynamical decomposition point,

in order to escape from a local optimal solution; and

d) approaching another stable equilibrium point of said nonlinear dynamical

system (4-10) satisfying said conditions (C2-1) and (C2-2) in the same

path-connected feasible component, via said dynamical decomposition

point.

34. -36. (Cancelled)

37. (Currently Amended) A hybrid search method for Phase II of claim 28, comprising the

following steps to find a local optimal solution of the constrained optimization problem:

(4.7).

a) given a feasible point x_0;

b) integrating a nonlinear dynamical system described-by-(4.10) that satisfies

required conditions (C2-1) and (C2-2)-from said initial point to obtain a

trajectory $\phi_t(x_0)$ for n time-steps, n is an integer, and recording the last

11

time-step point as the end point, and if it converges to a local optimal

solution, then stopping, otherwise, going to step (c);

c) monitoring a desired convergence performance criterion in an objective

function $C(x)$ ~~of (4.5)~~, and if the desired criterion is satisfied, then using

the end point of trajectory $\phi_t(x_0)$ as the initial point and going to step (b),

otherwise, going to step (d); and

d) applying an effective method from the said end point in step (b) to continue the

search process, and if it finds a feasible solution of ~~(4.7)~~ <u>the constrained</u>

<u>optimization problem</u> then stopping, otherwise, setting the end point of

trajectory $\phi_t(x_0)$ in step (b) as the initial point and going to step (b).

38. (Currently Amended) The method of claim 30, comprising the steps of:

a) choose a starting point;

b) initialization $j = 0$;

c) applying a hybrid search method for Phase I using said starting point to find a

feasible point in a (path-connected) feasible component, setting a point so

found as an initial point, and applying the hybrid search method for Phase

II to find a local optimal solution $x_{s,0}^j$;

d) starting from said initial point $x_{s,0}^j$, applying a numerical method for Phase II to

find all local optimal solutions in said feasible component and recording

them as the set V_s^j and set $V_s = V_s \cup V_s^j$;

e) for the local optimal solution $x_{s,0}^j$, defining a set of search vectors

$S_i^j, i = 1,2,...,k_j$;

f) if $i > k_j$, then going to step (l), otherwise, going to step (g);

12

g) for each search vector S_i^j, applying a reverse-time trajectory method to find a

point lying in an unstable equilibrium manifold of a system ~~(4.9)~~, and if a

point is found, letting it be denoted as $x_{d,j}^i$ and going to step (h), otherwise

setting $i = i + 1$ and going to step (f);

h) setting $x_{0,j}^i = x_s^j + (i + \varepsilon)(x_{d,j}^i - x_s^j)$, where ε is a small number and x_s^j is a local

optimal solution selected in step (e), and applying the hybrid search

method for Phase II using $x_{0,j}^i$ as the initial point to find a point lying in a

stable equilibrium manifold of system ~~(4.9)~~ satisfying required conditions

~~(C1-1) and (C1-2)~~, and letting the solution be denoted as $x_{s,j}^0$;

i) starting from said initial point $x_{s,j}^0$ and applying the hybrid search method of

step (b) of claim 28 to find a local optimal solution in said (path-

connected) feasible component $x_{s,0}^{j+1}$;

j) checking whether $x_{s,0}^{j+1}$ has been found before, i.e. $x_{s,0}^{j+1} \in V_s$?, and if it has been

bound before (i.e., the said feasible component has been visited before),

then setting $i = i + 1$ and going to step (f), otherwise setting

$V_s = V_s \cup \left\{ x_{s,0}^j \right\}$ and $V_{new}^{j+1} = V_{new}^{j+1} \cup \left\{ x_{s,0}^{j+1} \right\}$ and going to step (k);

k) starting from said initial point $x_{s,0}^{j+1}$, applying a numerical method for Phase II

to find all local optimal solutions in said feasible component and recording

them in the set V_s^{j+1}, set $V_s = V_s \cup V_s^{j+1}$ and setting $i = i + 1$ and going to

step (f);

l) examining the set of all newly computed local optimal solutions V_{new}^{j+1}, and if

V_{new}^{j+1} is empty, then going to the next step, otherwise, setting $j = j + 1$ and

going to step (e); and

13

m) identifying the global optimal solution from said set of local optimal solutions V_s by comparing their objective function values.

39. (Currently Amended) A numerical method for Phase II of claim 28, comprising the following steps to find all the local optimal solutions in said path-connected feasible component:

a) given an initial point lying in said path-connected feasible component;

b) initialization;

c) applying a hybrid search method for Phase II using the initial point to find a local optimal solution x_s^0, and setting $j = 0$, $V_s = \{x_s^0\}$, $V_{new.II}^j = \{x_s^0\}$ and $V_d = \{\phi\}$;

d) for each local optimal solution in V_{new}^j (i.e., x_s^j), performing steps (e) through (k);

e) defining a set of search vectors $S_i^j, i = 1,2,...,m_j$, and setting $i = 1$;

f) if $i > m_j$, then going to step (I), otherwise, applying a DDP search method, wherein a nonlinear dynamical system ~~(4.10)~~ satisfies <u>required</u> conditions ~~(C2-1) and (C2-2)~~ to find a corresponding dynamical decomposition point (DDP), and if a DDP is found, then going to step (g), otherwise, setting $i = i + 1$ and going to step (f);

g) letting the found DDP be denoted as $x_{d,i}^j$, checking whether it belongs to the set V_d, i.e., $x_{d,i}^j \in V_d$?, and if it does, then setting $i = i + 1$ and going to step (f), otherwise setting $V_d = V_d \cup \{x_{d,i}^j\}$ and going to step (h);

h) for the DDP $x_{d,i}^j$, performing steps (i) through (j) to find a corresponding adjacent local optimal solution;

14

i) setting $x_{0,i}^{j} = x_{s}^{j} + (1+\varepsilon)\left(x_{d,i}^{j} - x_{s}^{j}\right)$, where ε is a small number;

j) applying a hybrid search method using $x_{0,i}^{j}$ as the initial condition to find the corresponding local optimal solution, and letting it be denoted as $x_{s,j}^{i}$;

k) checking whether $x_{s,j}^{i}$ has been found before, i.e. $x_{s,j}^{i} \in V_{s}$?, and if it has already been found, then setting $i = i + 1$ and going to step (f), otherwise, setting $V_{s} = V_{s} \cup \left\{x_{s,j}^{i}\right\}$ and $V_{new}^{j+1} = V_{new}^{j+1} \cup \left\{x_{s,j}^{i}\right\}$ and setting $i = i + 1$ and going to step (f); and

l) examining the set of all newly computed local optimal solutions $V_{new,ll}^{j+1}$, and if $V_{new,ll}^{j+1}$ is non-empty, then setting $j = j + 1$ and proceeding to step (d), otherwise outputting all the local optimal solutions in said path-connected feasible component contained in set V_{s} and stopping the process.

40. (Previously Presented) A computer-implemented method for obtaining the global optimal solution of constrained nonlinear programming problems, comprising the steps of:

a) using a transformation technique to transform a constrained optimization problem into an unconstrained optimization problem, then applying the following steps to find the global optimal solution of the unconstrained optimization problem;

b) choosing a starting point;

c) apply a hybrid search method using said starting point to find a local optimal solution x_{s}^{0};

d) setting $j = 0$, $V_{s} = \left\{x_{s}^{0}\right\}$, $V_{new}^{j} = \left\{x_{s}^{0}\right\}$ and $V_{d} = \left\{\phi\right\}$;

e) setting $V_{new}^{j+1} = \left\{\phi\right\}$ and for each local optimal solution in V_{new}^{j} (i.e., x_{s}^{j}), performing steps (e) through (k);

15

f) defining a set of search vectors S_i^j, $i = 1,2, ..., m_j$, and setting $i = 1$;

g) if $i > m_j$, then going to step (l), otherwise, applying the DDP search method of claim 15 along the search vector S_i^j to find a corresponding dynamical decomposition point (DDP), and if a DDP is found, then going to step (g), otherwise, setting $i = i + 1$ and going to step (f);

h) letting the found DDP be denoted as $x_{d,i}^j$, checking whether it belongs to the set V_d, i.e., $x_{d,i}^j \in V_d$?, and if it does, then setting $i = i + 1$ and going to step (f), otherwise setting $V_d = V_d \cup \left\{ x_{d,i}^j \right\}$ and going to step (h);

i) for the DDP $x_{d,i}^j$, performing steps (i) through (j) to find a corresponding adjacent local optimal solution;

j) setting $x_{0,i}^j = x_s^j + (1 + \varepsilon)\left(x_{d,i}^j - x_s^j \right)$, where ε is a small number;

k) applying a hybrid search method using $x_{0,i}^j$ as the initial condition to find the corresponding local optimal solution, and letting it be denoted as $x_{s,j}^i$;

l) checking whether $x_{s,j}^i$ has been found before, i.e. $x_{s,j}^i \in V_s$?, and if it has already been found, then setting $i = i + 1$ and going to step (f), otherwise, setting $V_s = V_s \cup \left\{ x_{s,j}^i \right\}$ and $V_{new}^{j+1} = V_{new}^{j+1} \cup \left\{ x_{s,j}^i \right\}$ and setting $i = i + 1$ and going to step (f);

m) examining the set of all newly computed local optimal solutions, V_{new}^{j+1}, and if V_{new}^{j+1} is non-empty, then setting $j = j + 1$ and proceeding to step (d), otherwise proceeding to the next step; and

16

n) identifying the global optimal solution from said set of local optimal solutions

V_s by comparing their corresponding objective function values in set V_s;

and

o) displaying the global optimal solution.

41. (Cancelled)

17